

Creating Tables and Figures with \LaTeX

Ricky Patterson

bit.ly/LaTeXTables

5 Oct 2016

Outline

Introduction

Tables

- tabular environment
- multicolumn
- @-expressions

Figures

- picture environment
- Importing Graphics

References

Introduction

- ▶ \LaTeX is a document preparation system for high quality typesetting
- ▶ Figures and Tables are usually handled as "Floats" which, unlike words, cannot be broken across a line or a page
- ▶ \LaTeX allows you much control over the appearance and placement of tables and figures

L^AT_EX Tables

Tables are usually created within the tabular environment.

```
\begin{tabular}[pos]{table spec}
```

The table spec specifies the total number of columns, as well as the alignment for each column (l, c, or r). The option pos argument specifies the vertical location of the table with respect to the surrounding text (b, c, t). c is the default.

Keep in mind that \LaTeX is not a spreadsheet, and it is best to generate the values for a complex table externally, and then import the results into \LaTeX , adding the necessary formatting in an automated way if possible.

Keep in mind that \LaTeX is not a spreadsheet, and it is best to generate the values for a complex table externally, and then import the results into \LaTeX , adding the necessary formatting in an automated way if possible.

For instance: Each element in a \LaTeX table is separated from the adjacent column by an `&`. Unless you enjoy typing `&` symbols, you'll want to format the write statement of the code that is writing out your data to add a `&` symbol between columns to make it \LaTeX ready.

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1	cell 2	cell 3
header 5	cell 4	cell 5	cell 6

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1	cell 2	cell 3
header 5	cell 4	cell 5	cell 6

```
\begin{tabular}{c||c|c|c|}
& \textbf{header 1}& \textbf{header 2}& \textbf{header 4} \\ \hline
\textbf{header 4} & cell 1 & cell 2 & cell 3 \\ \hline
\textbf{header 5} & cell 4 & cell 5 & cell 6 \\ \hline
\end{tabular}
```


There are some additional symbols which can be used to describe table columns. Obviously | puts a vertical line between columns, and || puts two vertical lines.

There are some additional symbols which can be used to describe table columns. Obviously `|` puts a vertical line between columns, and `||` puts two vertical lines. In addition `p{'width'}` specifies a "paragraph" column, of width 'width'.

There are some additional symbols which can be used to describe table columns. Obviously `|` puts a vertical line between columns, and `||` puts two vertical lines. In addition `p{'width'}` specifies a "paragraph" column, of width 'width'. A non-p column does not wrap text within the cell, so if you have a lot of text in a cell, you may want to use `p` to control the appearance.

```
\begin{tabular}{| 1 | 1 | 1 | 1 |}
```

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine.

```
\begin{tabular}{| 1 | 1 | 1 | 1 |}
```

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine.

versus

```
\begin{tabular}{| 1 | 1 | 1 | p{5cm} |}
```

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine. However, the strong breeze will bring down the temperatures.

Use `\multicolumn` for cells which span more than one column:

```
\multicolumn{number cols}{align}{text}% align: l,c,r
```

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1 and 2		cell 3
header 5	cell 4	cell 5	cell 6

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1 and 2		cell 3
header 5	cell 4	cell 5	cell 6

```
\begin{tabular}{c||c|c|c|}  
& \textbf{header 1}&\textbf{header 2}& \textbf{header 4}\\ \hline\hline  
\textbf{header 4}& \multicolumn{2}{1}{cell 1 and 2}& cell  
3 \\  
\hline  
\textbf{header 5}& cell 4 & cell 5 & cell 6 \\  
\end{tabular}
```


L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1 and 2		cell 3
header 5	cell 4	cell 5	cell 6

```
\begin{tabular}{c||c|c|c|}  
& \textbf{header 1}&\textbf{header 2}& \textbf{header 4}\\ \hline\hline  
\textbf{header 4}& \multicolumn{2}{1|}{cell 1 and 2}& cell 3  
3 \\  
\hline  
\textbf{header 5}& cell 4 & cell 5 & cell 6 \\  
\end{tabular}
```

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1 and 2		cell 3
header 5	cell 4	cell 5	cell 6

L^AT_EX Tables

	header 1	header 2	header 4
header 4	cell 1 and 2		cell 3
header 5	cell 4	cell 5	cell 6

```
\begin{tabular}{c||c|c|c|}
& \textbf{header 1}&\textbf{header 2}& \textbf{header 4}\\
\hline\hline
\textbf{header 4}& \multicolumn{2}{1}{cell 1 and 2}& cell 3 \\
\hline
\textbf{header 5}& cell 4 & cell 5 & cell 6 \\
\hline
\end{tabular}
```

Note: `\multicolumn{1}{align}{text element}` can be used to simply alter the default column centering for that particular text element.

A Column Separator can be specified using the **@-expression**.

A Column Separator can be specified using the **@-expression**.
By default, columns are separated with tabs (or vertical lines).
However, using an @-expression, columns can be separated
with decimal points or dashes (or any other character).

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

left-aligned

3.14159

16.2

123.456

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

left-aligned	centered
3.14159	3.14159
16.2	16.2
123.456	123.456

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

left-aligned	centered	right-aligned
3.14159	3.14159	3.14159
16.2	16.2	16.2
123.456	123.456	123.456

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

left-aligned	centered	right-aligned	@-aligned
3.14159	3.14159	3.14159	3.14159
16.2	16.2	16.2	16.2
123.456	123.456	123.456	123.456

A Column Separator can be specified using the **@-expression**. By default, columns are separated with tabs (or vertical lines). However, using an @-expression, columns can be separated with decimal points or dashes (or any other character). This can be very useful if you wish to align a list of numbers on the decimal point:

left-aligned	centered	right-aligned	@-aligned
3.14159	3.14159	3.14159	3.14159
16.2	16.2	16.2	16.2
123.456	123.456	123.456	123.456

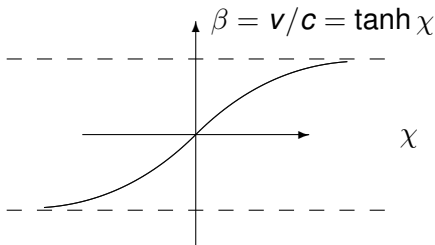
```
\begin{tabular}{r@{.}l}
```

```
3 & 14159 \\
```

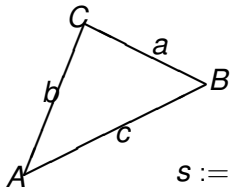
```
16 & 2 \\
```

```
123 & 456 \\
```

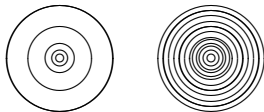
It is possible to create figures within \LaTeX (usually within the picture environment), and there are packages available which simplify the creation of certain types of figures (like the tikz package for flowcharts, or pstricks which enhances the picture environment). This has a couple of advantages: the document is self-contained and small, and you have access to the fonts and math notation within \LaTeX .



$$F = \sqrt{s(s-a)(s-b)(s-c)}$$



$$s := \frac{a + b + c}{2}$$



However, most complex figures you use will probably be produced in other programs (it's just easier), and the resulting PDF or EPS file is imported into the \LaTeX file. You will need to use the `graphicx` package in order to import graphics.

However, most complex figures you use will probably be produced in other programs (it's just easier), and the resulting PDF or EPS file is imported into the \LaTeX file. You will need to use the `graphicx` package in order to import graphics.

Keep in mind: If you are compiling directly with latex command, you can only use EPS files. However, most compilation is now done with `pdflatex`, which can handle additional formats, including JPG (good for photos), PNG (lossless, good for diagrams and screenshots), PDF (can be used for vector graphics), and EPS (but you may have to load the `epstopdf` package).

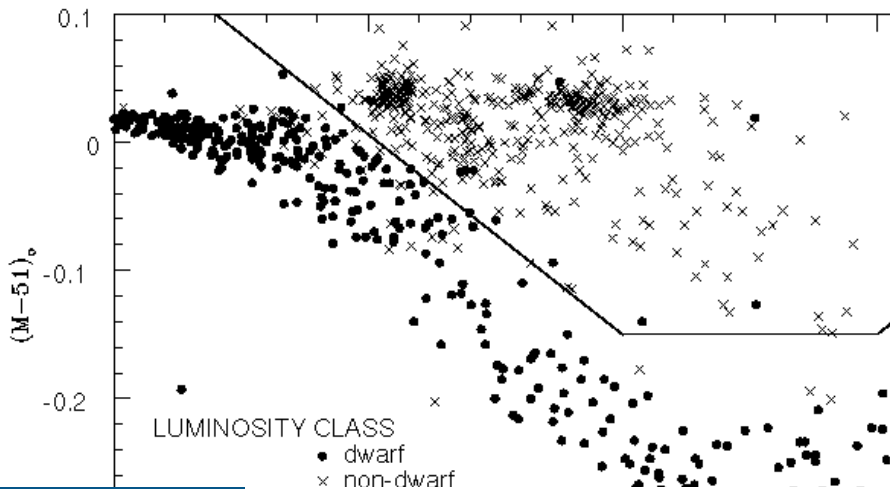
Using the `graphicx` environment, you will use the `includegraphics` command to import the figure, while scaling, rotating and offsetting it to place it exactly where you want it.

```
\includegraphics[attr1=val1, ... attrn=valn]{image}
```

Possible attributes include: `width`, `height`, `scale`, `angle`, `resolution`, and `keepaspectratio`, as well as `trim`, `clip`, and `page`.

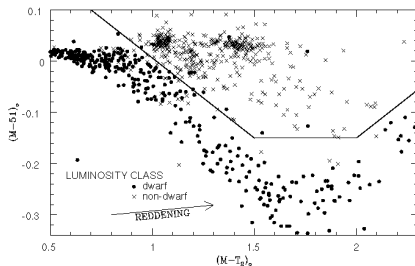
Let's try importing a figure.

```
\includegraphics{1.eps}
```



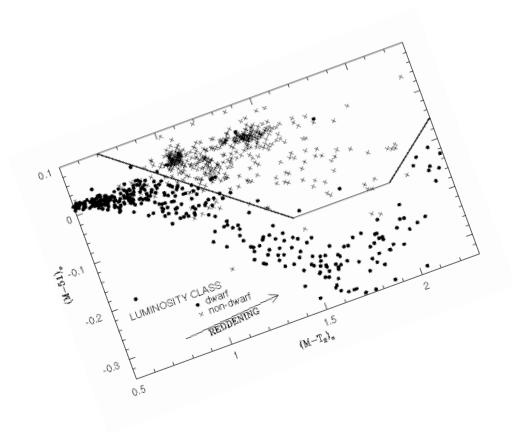
Set the scale to 40%, so it fits better on the page:

```
\includegraphics[scale=0.4]{14.eps}
```



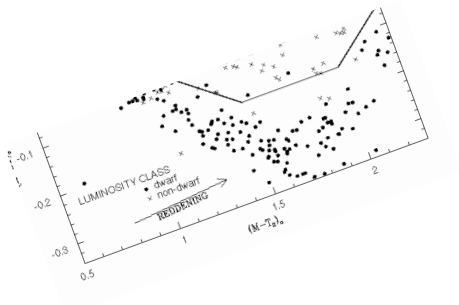
Now let's rotate it, just for fun:

```
\includegraphics[scale=0.4, angle=20]{14.eps}
```



And trim it a bit (note that trim is performed prior to rotation in this case):

```
\includegraphics[trim= 3mm 60mm 1mm 10cm, scale=0.4, angle=
```



References

- ▶ CTAN (Comprehensive T_EX Archive Network): ctan.org
- ▶ TUG (T_EX Users Group): tug.org
- ▶ The L^AT_EX Companion (Mittelbach and Goossens, 2004)
- ▶ A Guide to L^AT_EX: Document Preparation for Beginners and Advanced Users (Kopka and Daly, 1999)
- ▶ The Not So Short Intro to L^AT_EX (Oetiker, updated July 2015 → lshort.pdf)
- ▶ L^AT_EX WikiBook <https://en.wikibooks.org/wiki/LaTeX/>
- ▶ This presentation (code and PDF): <http://bit.ly/LaTeXTables>
- ▶ Materials from previous L^AT_EX workshops: <http://data.library.virginia.edu/statlab/past-workshops/>
- ▶ Overleaf.com Templates and Intro Guides.